ICSI 518 Software Engineering

Software Requirement Specification (SRS)

CommunityHub360

Table of Contents

[The Purpose of the Product 3](#_Toc161006654)

[1.a Project Background: 3](#_Toc161006655)

[1.b Goals: 4](#_Toc161006656)

[1.c Scope of the project: 4](#_Toc161006657)

[2. Stakeholders 5](#_Toc161006658)

[2a. Description of users - Customers 5](#_Toc161006659)

[2b. Description of characteristics of the Users: 6](#_Toc161006660)

[3. Mandated Constraints 7](#_Toc161006661)

[3a. Solution Constraints: 7](#_Toc161006662)

[3b. Implementation Environment 7](#_Toc161006663)

[3c. Partner or Collaborative Applications 7](#_Toc161006664)

[3d. Off-the-Shell-Software 8](#_Toc161006665)

[3e. Schedule 8](#_Toc161006666)

[4. Naming Conventions and Terminology: 12](#_Toc161006667)

[5. Assumptions 13](#_Toc161006668)

[6. Scope 14](#_Toc161006669)

[6a) Use case Diagram 14](#_Toc161006670)

[6b) Use case Tables 15](#_Toc161006671)

[7. Functional Requirements: 17](#_Toc161006672)

[7a). Complete Set of Functional Requirements: 17](#_Toc161006673)

[7b) Traceability Matrix 25](#_Toc161006674)

[8. Non-Functional Requirements 27](#_Toc161006675)

[8.1. Product Requirements 27](#_Toc161006676)

[8.2. Organizational Requirements 27](#_Toc161006677)

[8.3. External requirements 27](#_Toc161006678)

# The Purpose of the Product

## 1.a Project Background:

CommunityHub360 was created to address the inefficiencies that plague the management of Apartment complexes and Housing Societies (HSCs). Currently, the existing systems have a lack of centralised communication channels. This leads to a lack of coordination among different stakeholders, such as the administrator, the residents, the security team, and the maintenance team. This lack of coordination often results in delays, miscommunication, and substandard management practices, affecting the quality of life in these communities.

With an advanced web platform, CommunityHub360 seeks to completely transform the way these communities are run. Simplifying the numerous procedures involved—from event planning and tenant management to maintenance requests and security measures—is the major goal. The application of contemporary technology solutions helps to close the communication gap and promote smooth cooperation between the various stakeholders engaged in the community's daily operations.

The primary goal of CommunityHub360 is to improve the general effectiveness and communication within the community. The platform intends to promote more efficient workflows and decision-making processes by offering a central portal where residents can readily communicate with administrators, security teams, and maintenance personnel. CommunityHub360 aims to establish an atmosphere where all community members feel connected, informed, and empowered to contribute to the overall well-being of the community through user-friendly interfaces and cutting-edge technologies.

All things considered, CommunityHub360 is a revolutionary way to manage housing societies and apartments. Through addressing the underlying reasons of inefficiencies and communication failures, the platform aims to improve living conditions in these communities and promote a sense of community, security, and cohesion among the occupants. CommunityHub360 aims to raise the bar for community management through constant iteration and refinement, opening the door for a more peaceful and satisfying living environment for all parties concerned.

## 1.b Goals:

By establishing a centralized platform, CommunityHub360 seeks to address the inefficiencies in the management of apartment buildings and housing societies. The project intends to introduce

advanced security measures, streamline complex procedures for better task management, and

enhance communication between maintenance, security, and administration teams as well as residents. The platform's objective is to deliver a seamless user experience by

encouraging creativity, engaging the community, and making use of cutting-edge tools like AWS, MongoDB, HTML, CSS, JS, and Bootstrap. Moreover, putting in place a robust feedback system ensures resident satisfaction and a well-maintained environment while facilitating continuous improvements.

The target of CommunityHub360 is to become a central point of contact and information by providing intuitive user interfaces and cutting-edge capabilities, including brand collaborations, online voting, and community news feeds. The primary goal of the project is to encourage community involvement in activities, enhance feedback, and keep everyone updated with a view to cultivate a vibrant and active community. The CommunityHub360 improves communication, streamline operations, and develop an innovative technology platform that enhances residents' overall satisfaction and well-being.

## 1.c Scope of the project:

The purpose of CommunityHub360 is to simplify the management of housing and housing associations by providing various functions. It contributes to resident information by making it easy to add or update information. The platform is a place for community events, so it's easy for residents to stay informed and participate. Residents can also submit and track maintenance requests, ensuring a quick response to issues. Security measures include functions such as managing visitors and providing parking information to improve security.

# 2. Stakeholders

## 

## 2a. Description of users - Customers

The users of our web page are:

Administrator:

Monitor community operations, manage tenant information and coordinate events to ensure effective

community management..

Residents:

Access community information, attend events, submit service requests and stay up to date with

notifications for an enriched life experience..

Security Team:

Track parking information, manage visitor information and ensure overall community safety..

Maintenance Team:

Process maintenance requests, review apartments and maintain community infrastructure to ensure a well-maintained living environment.

## 2b. Description of characteristics of the Users:

Administrator:

* admin can login directly to web page
* admin has authority to accept the tenants signup request
* admin gets involved authorities
* admin manages community affairs
* admin ensuring effective communication and coordination.
* Can change password

Residents:

* Has to signup as a new tanent
* Tanent can view the surroundings through web page as announcements
* Gets involved with the parties and meetings
* Can raise a request to maintainence team for any house hold issues
* Can raise a request to parking team for the parking issues
* Can raise request to admin for room/hall booking issues
* Can change password

Security Team:

* Can login through the details provide by admin
* Can acces the issues raised by tanent
* maintaining community safety
* can change password

Maintenance Team:

* Has to login by the details provided by admin
* Can access the issues regarding house hold raised by tanents
* ensuring the community&#39;s physical well-being.
* Can change password

# 3. Mandated Constraints

## 3a. Solution Constraints:

The product must be capable of functioning on various web browsers, including Google Chrome, Safari, Opera, Firefox, and others. It should also be able to operate on both mobile phones and personal computers.

The following technologies were used in building the product:

**React JS**: a JavaScript library used for building user interfaces.

**Nest JS:** a Node.js framework that provides TypeScript support and more built-in features.

**CSS**: a CSS framework that offers rapid development, customizability, responsive design, consistency, and improved performance.

**Bootstrap**:

**AWS:** a cloud computing platform that provides a wide range of services and tools for building, deploying, and managing applications.

**MongoDB**: a NoSQL database that provides flexibility, scalability, developer productivity, and availability.

**Git**: a version control system that provides version control, collaboration, branching and merging, code review, and backup and recovery.

**GitHub:** a web-based platform that provides version control, collaboration, issue tracking, pull requests, continuous integration and deployment, and community and open-source support.

**Postman**: a tool used in application development to simplify the process of testing and debugging APIs.

## 3b. Implementation Environment

* Product will be added to the current network of computers.
* For the project's dynamic web page creation and component reuse, ReactJS will be utilized while MongoDB will serve as the data storage system. Hashing, on the other hand, will be used for authentication and AWS is used for hosting purposes.

## 3c. Partner or Collaborative Applications

* The product will not rely on any third-party or cooperative application.

## 3d. Off-the-Shell-Software

* The product will not utilize any off-the-shelf software.

### 3e. Schedule

|  |  |  |
| --- | --- | --- |
| **Tasks** | **Duration** | **Estimate points/Effort**  **(days)** |
| Created an environmental setup in AWS for the project | 01/30/2024 – 02/01/2024 | 1 |
| Created the backend environmental setup | 02/01/2023 – 02/04/2024 | 3 |
| Created an authentication for Admin, Tenant modules | 02/04/2024 – 02/06/2024 | 2 |
| Created an Manage Tenant page  in the admin module | 02/06/2024 – 02/09/2024 | 3 |
| Created an announcements view page in the admin module | 02/09/2024 – 02/11/2024 | 3 |
| Created an Add Announcement page in the admin module | 02/11/2024 – 02/12/2024 | 2 |
| Created database schemas and tables for admin module | 02/12/2024 – 02/14/2024 | 2 |
| Created an view announcements  page  in the Tenant  module | 02/14/2024 – 02/17/2024 | 3 |
| Created an Maintenance request  page  in the Tenant  module | 02/17/2024 – 02/19/2024 | 2 |
| Created an Guestparking page  in the Tenant  module | 02/19/2024 – 02/19/2024 | 1 |
| Created an Booking request page and about page  in the Tenant  module | 02/19/2024 – 02/22/2024 | 3 |
| Created an about page  in the Tenant  module | 02/22/2024 – 02/25/2024 | 2 |
| Created database schemas and tables for a Tenant module | 02/25/2024 – 02/28/2024 | 3 |
| Created an Manage Tenant Parking in the Security Model | 02/29/2024 – 03/02/2024 | 4 |
| Created an Manage Guest Parking in the Security Model | 03/02/2024 – 03/03/2024 | 1 |
| Created database schemas and tables for a Security  module | 03/03/2024 – 03/05/2024 | 3 |
| Created an Unresolved Maintenance request list page in the Maintenance Module | 03/05/2024 – 03/07/2024 | 3 |
| Created a Resolved Maintenance request list page in theMaintenance Module | 03/07/2024 – 03/09/2024 | 4 |
| Created database schemas and tables for a Maintenance module | 03/09/2024 – 03/10/2024 | 1 |
| Will create the B$S Forum page in the Tenant module | 03/19/2024 – 03/16/2024 | 3 |
| Will create the Buy and view  products page in the Tenant module | 03/16/2024 – 03/21/2024 | 3 |
| Will create the Sell products page in the Tenant module | 03/21/2024 – 03/28/2024 | 2 |
| Will create the Booking requests page  in the Security  module | 03/28/2024 – 04/02/2024 | 3 |
| Will create database schemas and tables for the new security model and tenant module | 04/02/2024 – 04/08/2024 | 6 |
| End to end testing of admin module | 04/09/2024 – 04/13/2024 | 5 |
| End to end testing of Tenant  module | 04/13/2024 – 04/19/2024 | 4 |
| End to end testing of  Maintenance  module | 04/19/2024 – 04/24/2024 | 6 |
| End to end testing of  Security module | 04/24/2024 – 04/25/2024 | 1 |
| Deployment of project to AWS | 04/25/2024 – 04/29/2024 | 4 |

# 4. Naming Conventions and Terminology:

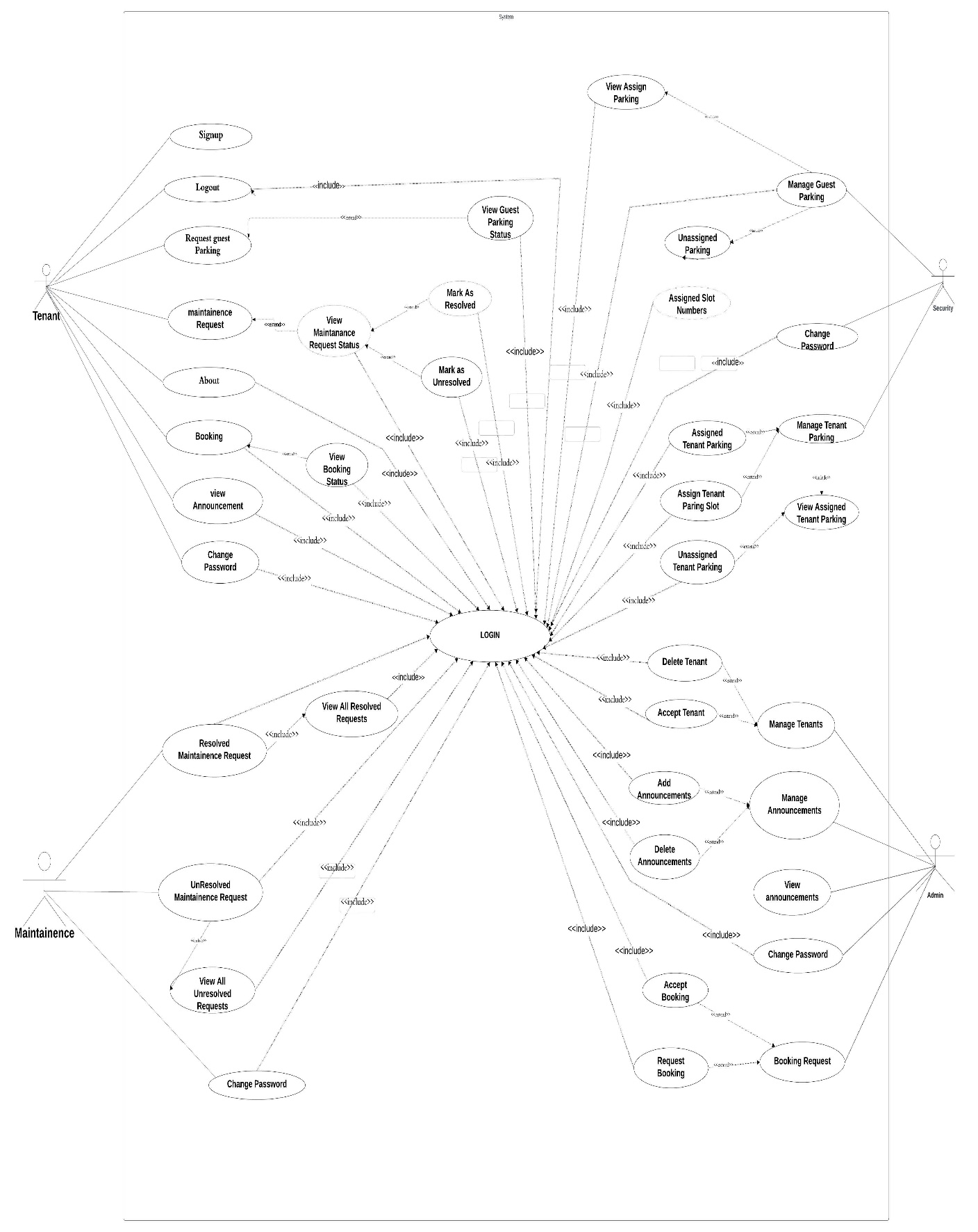
|  |  |
| --- | --- |
| **Term** | **Description** |
| API | Application programming interface is a set of rules and protocols used to create software applications. |
| Authentication | The process of authenticating a user's or system's identification before granting access to protected resources. |
| User Interface (UI) | A software application's visual elements and interactions that allow users to interact with the application. |
| Database | A collection of data that has been structured in such a way that it can be stored, retrieved, and managed efficiently. |
| Server | A computer or system that, via a network, delivers services or resources to other computers or systems. |
| Admin Module | A module in a software application that allows system administrators to manage the application, data, and users by providing access and functionality. |
| Tenant Module | A module that allows tenants to edit their information, and access the announcements posted by admin, allows them to raise maintenance requests and parking requests. |
| Maintenance module | This module allows access to maintenance requests raised by tenants and allows appointment scheduling to fix the issues |
| Security Module | This module allows access to assign parking slots to tenants and guests on request basis |
| Schema | A blueprint or framework that governs how data is organized and related in a database or application. |

# 5. Assumptions

* It is important for all customers to have a good understanding of how to use the web application.
* It is important that every customer can use either a personal computer or a smartphone.

# 6. Scope

## 6a) Use case Diagram



### 6b) Use case Tables

|  |  |  |  |
| --- | --- | --- | --- |
| **Use**  **Case ID** | **Use Case**  **Name** | **Actor(s)** | **Input or Output** |
| 001 | Signup | Tenant | All the actors of this application must be able to complete the Register successfully by giving all the valid details of their respective role. |
| 002 | Login | Admin, Tenant, Security | I/P: All actors should give their valid credentials as input to log into the system.  O/P: The Login should be successful for the valid credentials. |
| 003 | change password | Admin, Tenant, Security | All Actors  can change the password for the security purpose |
| A01 | Manage Tenants | Admin | Admin can accept or reject the tenants |
| A02 | Manage Announcements | Admin | admin can add the announcements |
| A03 | View Announcements | Admin | admin  can view the announcements |
| T04 | View Announcements | Tenant | Tenant can view the announcements |
| T01 | Maintenance Request | Tenant | Tenants can raise the maintenance request |
| T02 | Guest parking status | Tenant | Tenants can raise the guest parking  request |
| T03 | Booking status | Tenant | Tenants can raise the booking request |
| T04 | About | Tenant | Tenants can view the about |
| S01 | Manage Tenant parking | Security | Security can assign  tenant parking slot |
| S02 | Manage Guest parking | Security | Security can assign the guest  parking slot |
| M01 | Resolved maintenance request list | Maintenance | Maintenance team  can view the resolved request list |
| M02 | UnResolved maintenance request list | Maintenance | Maintenance team  can view the Unresolved request list |

# 7. Functional Requirements:

## 7a). Complete Set of Functional Requirements:

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 1 |  | Event/Use Case #:         UC01 |
| Description: The Administrator needs to provide their Username and Password to sign in to the system. | | |
| Rationale: The login system should ensure that only authorized administrators can access the system by requiring them to provide correct login credentials, such as a username and password, before granting access. | | |
|  | | |
| Fit Criterion:   * Preconditions   The admin must be registered first to sign in to the system and access it.  The system must treat usernames as case-sensitive entities and enforce a policy where duplicate usernames, considering case sensitivity, are not permitted within the database.   * Postconditions   The system should verify that the Admin’s login information (username and password) matches the stored records, ensuring that only authorized individuals can access the system. In case the Admin's login details are incorrect during the authentication process, the system should display a message to the Admin informing them that the authentication has failed due to invalid credentials.  Upon successfully authenticating, the user should be directed to the admin UI of the application | | |
| Dependencies: Accessing the system requires prior registration. Only individuals authorized as administrators can log in to get access to the admin dashboard. | | |
| Supporting Materials: | | |

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 2 |  | Event/Use Case #:         UC02 |
| Description: Admin must be able to manage tenant information. | | |
| Rationale: The system shall ensure that only successfully logged-in admin can perform this action. | | |
|  | | |
| Fit Criterion:   * Preconditions   Tenants should be registered or trying to register into the community so that their their name, house number, mobile number to be displayed in the manage tenants section.   * Postconditions   By clicking accept/delete, the system shall add/delete the tenant information from the community database. | | |
| Dependencies: Accessing this feature requires an admin login. Only the individuals authorized as Admin can manage tenant information. | | |
| Supporting Materials: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 3 |  | Event/Use Case #:         UC03 |
| Description: Admin must be able to post announcements. | | |
| Rationale: The system shall ensure that only successfully logged-in admin can perform this action. | | |
|  | | |
| Fit Criterion:   * Preconditions   No new announcement has been posted by the admin. There can be existing announcements.   * Postconditions   Upon successfully adding the announcement by entering a title, and description and clicking on add announcement, the new announcement must be posted and the tenants must be able to see it. | | |
| Dependencies: Accessing this feature requires an admin login. Only the individuals authorized as Admin can post announcements | | |
| Supporting Materials: | | |
|  | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 4 |  | Event/Use Case #:         UC04 |
| Description: The admin must be able to view announcements they have posted on the admin dashboard. | | |
| Rationale: The system shall ensure that only successfully logged-in admin can perform this action. | | |
|  | | |
| Fit Criterion:   * Preconditions   An announcement must be posted first.   * Postconditions   The admin must be able to view an announcement with a given title, description, time, and date on which the announcement has been posted. The system shall display the recent announcement which has been made on top of the announcements list. The system shall update this announcement on the tenant view announcement section on the tenant dashboard. | | |
| Dependencies: Accessing this feature requires an admin login. Only the individuals authorized as Admin can view announcements on the admin dashboard. | | |
| Supporting Materials: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 5 |  | Event/Use Case #:         UC05 |
| Description: The admin must be able to delete announcements they have posted on the admin dashboard. | | |
| Rationale: The system shall ensure that only successfully logged-in admin can perform this action. | | |
|  | | |
| Fit Criterion:   * Preconditions   Announcements must have been posted.   * Postconditions   The admin must be able to delete the announcement they have made by clicking on delete icon beside the announcement in the existing announcements section. | | |
| Dependencies: Accessing this feature requires an admin login. Only the individuals authorized as Admin can view announcements on the admin dashboard. | | |
| Supporting Materials: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 6 |  | Event/Use Case #:         UC06 |
| Description: The admin must be able to change their login password | | |
| Rationale: The system shall ensure that only successfully logged-in admin can perform this action. | | |
|  | | |
| Fit Criterion:   * Preconditions   The Admin must be logged in to perform this action.   * Postconditions   The admin after clicking on change password section on the admin dashboard and entering his new password and confirming it, the system shall show a pop-up message saying the password changed successfully and update it in the database. The admin after logging out should enter his new password to log in again. | | |
| Dependencies: Accessing this feature requires an admin login. Only the individuals authorized as Admin can change their passwords. | | |
| Supporting Materials: | | |
|  | | |

**Tenant**

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 7 |  | Event/Use Case #:         UC07 |
| Description: Tenants must register themselves by providing required details which are first name, last name, username, house number, phone number, and their password) | | |
| Rationale: Tenants must sign up in order to log in to the system and provide required information. | | |
|  | | |
| Fit Criterion:   * Preconditions   The user does not have an account in the system.   * Postconditions   Following successful registration, the system shall show a pop-up message to the user saying registration is successful and send this registration request to the admin where they can accept or deny the registration request. The user can only log in as a tenant after the admin accepts the registration request. | | |
| Dependencies: The username should not match with any of the usernames in the database and password must contain atleast one capital letter(A-Z), one number(0-9), one special character (#,!,&,%), system shall mask the typed, password and the password must be minimum of 8 characters. | | |
| Supporting Materials: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 8 |  | Event/Use Case #:         UC08 |
| Description: The Tenants needs to provide their Username and Password to sign in to the system. | | |
| Rationale: The login system should ensure that only authorized tenants can access the system by requiring them to provide correct login credentials, such as a username and password, before granting access. | | |
|  | | |
| Fit Criterion:   * Preconditions   The tenant must be registered first to sign in to the system and access it.  The system must treat usernames as case-sensitive entities and enforce a policy where duplicate usernames, considering case sensitivity, are not permitted within the database.   * Postconditions   The system should verify that the user's login information (username and password) matches the stored records, ensuring that only authorized individuals can access the system. In case the Tenant's login details are incorrect during the authentication process, the system should display a message to the tenant informing them that the authentication has failed due to invalid credentials.  Upon successfully authenticating, the user should be directed to the tenant UI of the application. | | |
| Dependencies: Accessing the system requires prior registration. Only individuals authorized as tenants can log in to get access to the tenant dashboard. | | |
| Supporting Materials: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 9 |  | Event/Use Case #:         UC09 |
| Description: Tenants must be able to view the announcements posted by the admin. | | |
| Rationale: The system shall ensure that only successfully logged-in tenants can perform this action on the tenant dashboard. | | |
|  | | |
| Fit Criterion:   * Preconditions   New or existing announcements must be posted by the admin.   * Postconditions   After the admin has posted an announcement, the system shall display the posted announcement on the view announcements section of tenant dashboard. | | |
| Dependencies: Accessing this feature requires a tenant login. Only the individuals authorized as tenants can view announcements on the tenant dashboard. | | |
| Supporting Materials: | | |

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 10 |  | Event/Use Case #:         UC10 |
| Description: Tenants must be able to submit new maintenance requests. | | |
| Rationale: The system shall ensure that only successfully logged-in tenants can perform this action on the tenant dashboard. | | |
|  | | |
| Fit Criterion:   * Preconditions   No new maintenance requests have been submitted by the tenant.   * Postconditions   After the tenant submits the maintenance request with the appropriate priority, maintenance category, full description of the problem, and the availability in which the request can be resolved, the system shall pass on this information to the security team who deal with the issues raised | | |
| Dependencies: Accessing this feature requires a tenant login. Only the individuals authorized as tenants can submit new maintenance requests on the tenant dashboard. | | |
| Supporting Materials: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 11 |  | Event/Use Case #:         UC11 |
| Description: Tenants must be able to view the status of the submitted maintenance requests. | | |
| Rationale: The system shall ensure that only successfully logged-in tenants can perform this action on the tenant dashboard. | | |
|  | | |
| Fit Criterion:   * Preconditions   New or existing maintenance requests should have been submitted by the tenant.   * Postconditions   The tenants can view the new or existing maintenance requests by clicking on my maintenance requests on the maintenance request section with the request status which is marked as resolved (if tenants click on resolved) and not resolved otherwise. | | |
| Dependencies: Accessing this feature requires a tenant login. Only the individuals authorized as tenants can view the status of the maintenance requests on the tenant dashboard. | | |
| Supporting Materials: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 12 |  | Event/Use Case #:         UC12 |
| Description: Tenants must be able to request a new guest parking slot for their guests. | | |
| Rationale: The system shall ensure that only successfully logged-in tenants can perform this action on the tenant dashboard. | | |
|  | | |
| Fit Criterion:   * Preconditions   No new guest parking requests have been submitted by the tenant.   * Postconditions   After the tenant submits a service request with the required details (vehicle type, number, state from which it is from, guests’ name, number of days their vehicle needs the parking space), the system shall submit this request to the Security team whose role (one of the roles) is to assign the available parking slot. | | |
| Dependencies: Accessing this feature requires a tenant login. Only the individuals authorized as tenants can post guest parking requests on the tenant dashboard. | | |
| Supporting Materials: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 13 |  | Event/Use Case #:         UC13 |
| Description: Tenants must be able to view guest parking slots status for their guests. | | |
| Rationale: The system shall ensure that only successfully logged-in tenants can perform this action on the tenant dashboard. | | |
|  | | |
| Fit Criterion:   * Preconditions   New or existing guest parking requests should have been submitted by the tenant.   * Postconditions   After the security team assigns the available parking slot to the concerning guest parking requests, the tenants can view the status by clicking on view my booking status section in the guest parking section. | | |
| Dependencies: Accessing this feature requires a tenant login. Only the individuals authorized as tenants can view the status of guest parking requests on the tenant dashboard. | | |
| Supporting Materials: | | |

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 14 |  | Event/Use Case #:         UC14 |
| Description: Tenants must be able to book a room or hall inside the community. | | |
| Rationale: The system shall ensure that only successfully logged-in tenants can perform this action on the tenant dashboard. | | |
|  | | |
| Fit Criterion:   * Preconditions   Tenants can book any available room or hall only.   * Postconditions   After the tenant submits a booking request with the required details (booking type, event name, number of guests, and number of days of booking), the system shall submit this request to the Admin whose role (one of the roles) is to assign the available room or hall inside the community. | | |
| Dependencies: Accessing this feature requires a tenant login. Only the individuals authorized as tenants can post a booking request on the tenant dashboard. | | |
| Supporting Materials: | | |

|  |  |  |
| --- | --- | --- |
| **Requirement Shell** | | |
| Requirement #: 15 |  | Event/Use Case #:         UC15 |
| Description: The tenant must be able to change their login password | | |
| Rationale: The system shall ensure that only successfully logged-in tenant can perform this action. | | |
|  | | |
| Fit Criterion:   * Preconditions   The tenant must be logged in to perform this action.   * Postconditions   The tenant after clicking on change password section on the tenant dashboard and entering his new password and confirming it, the system shall show a pop-up message saying the password changed successfully and update it in the database. The tenant after logging out should enter his new password to log in again. | | |
| Dependencies: Accessing this feature requires a tenant login. Only the individuals authorized as tenants can change their password. | | |
| Supporting Materials: | | |
|  | | |
|  | | |
|  | | |

## 

## 7b) Traceability Matrix

|  |  |  |
| --- | --- | --- |
| **User Story** | **Description/notes** | **Requirements** |
| U1: As an admin, I should provide my Username & Password to sign in to system. | Admin’s must provide login details to use the system. | R1 |
| U2: As an Admin, I want to manage tenant information | Admins should be able to manage tenant information, i.e., updating tenant details from the community database records. | R2 |
| U3: As an Admin, I must be able to post announcements | As an Admin, I must be able to post announcements. | R3 |
| U4: As an Admin, I should be able to see announcements I have posted. | Admins must be able to view announcements they posted in admin dashboard. | R4 |
| U5: As an admin, I should be able to delete the announcements I have posted | As a admin, I should be able to delete the announcements I have posted. | R5 |
| U6: As an admin I should be able to change my password. | Admins should be able to update their passwords. | R6 |
| U7: As a Tenant, I should provide my Username and Password to sign in to the system. | Tenants must provide their login details (Username and Password) to access the system. | R7 |
| U8: As a Tenant, I should be able to view the announcements posted by the admin. | As a tenant I must be able to view announcements that, I have a posted on the dashboard. | R8 |
| U9: As a Tenant, I should be able to submit new maintenance requests. | As a Tenants I must be able to submit new maintenance requests. | R9 |
| U10: As a Tenant, I must be able to view the status of my submitted maintenance requests. | As a Tenants, I should be able to check the status of their submitted maintenance requests. | R10 |
| U11: As a Tenant, I must be able to request a new guest parking slot for my guests. | As a Tenants, I should be able to request a new guest parking slot for their guests. | R11 |
| U12: As a Tenant, I must be able to request a parking slot status for my guests. | As a Tenants, I should be able to view guest parking slot status for their guests. | R12 |
| U13: As a Tenant, I must be able to book a room or hall inside the community. | As a Tenants, I should be able to book a room or hall inside the community. | R13 |
| U14: As a Tenant, I must be able to change my login password. | All Tenants should be able to change my login password. | R14 |

# 8. Non-Functional Requirements

## 8.1. Product Requirements

**8.1.a Usability:** The interface of the application is designed to be intuitive and userfriendly, enabling users to place orders effortlessly by simply logging in, without encountering any issues.

**8.1.b Scalability**: With a growing user base, the app has been carefully crafted to accommodate an increasing number of users. It has been equipped with the ability to scale up or down as needed and efficiently manage the storage of data and incoming traffic

**8.1.c Adaptability**: The CommunityHub360 application should be adaptable to different device types, screen sizes, and operating systems to ensure seamless user experience and wider accessibility.

**8.1.d Reliability:** The application offers a seamless experience to customers, ensuring greater reliability. Issues raised through the app are quickly processed, minimizing response times and enhancing the overall experience.

## 8.2. Organizational Requirements

**8.2.a Implementation:** Hashing will be utilized for password authentication, while data storage and retrieval will be handled through MongoDB in the web application. The development of dynamic web pages was accomplished using ReactJS. The web application is currently being hosted on AWS.

## 8.3. External requirements

**8.3.a Security:** To ensure the safety and privacy of user information, the web application has implemented strong security measures and adheres to relevant privacy regulations. These measures are in place to prevent any unauthorized access, misuse, or compromise of user data.

**8.3.b Maintainability**:

This application is very flexible to maintain all the data related to the Admin, tenant, maintenance, and security team in the database.